Package 'rainbow'

February 15, 2012

Type Package

Title Rainbow plots, bagplots and boxplots for functional data

Version 2.8

Date 2012-01-10

Depends R (>= 2.10.0), hdrcde, MASS, pcaPP, cluster, colorspace

Author Han Lin Shang and Rob J Hyndman

Maintainer Han Lin Shang <HanLin.Shang@monash.edu.au>

Description Functions and datasets for functional data display and outlier detection.

License GPL (>= 2)

LazyData yes

LazyLoad yes

LazyDataCompression xz

URL http://monashforecasting.com/index.php?title=R_packages

Repository CRAN

Date/Publication 2012-01-10 08:30:48

R topics documented:

`rainbow-package` *Rainbow plots, bagplots and boxplots for functional data*

Description

This package computes the rainbow plots, bagplots and boxplots for functional data. The latter two can also be used to identify outliers, which have either the lowest depth or the lowest density.

Author(s)

Han Lin Shang and Rob J Hyndman

Maintainer: Han Lin Shang <HanLin.Shang@monash.edu>

References

R. J. Hyndman and H. L. Shang. (2008) "Bagplots, boxplots and outlier detection for functional data", in S. Dabo-Niang and F. Ferraty, eds, ‘Functional and Operatorial Statistics’, Springer, Heidelberg, pp. 201-207.

R. J. Hyndman and H. L. Shang. (2010) "Rainbow plots, bagplots, and boxplots for functional data", *Journal of Computational and Graphical Statistics*, **19**(1), 29-45.

`Australiafertility` *Australian fertility data*

Description

Age-specific fertility rates between ages 15 and 49 in Australia from 1921 to 2006.

The age-specific fertility rates can be smoothed using a weighted median smoothing B-splines, constrained to be concave.

Usage

```
data(Australiafertility)
data(Australiasmoothfertility)
```

Format

An object of class `fts`.

Details

Australian fertility rates and populations (1921-2006) for age groups (15-49) were obtained from the Australian Bureau of Statistics (Cat.No.3105.0.65.001, Table 38). These are defined as the number of live births during the calendar year, according to the age of the mother, per 1000 of the female resident population of the same age at 30 June.

Australiasmoothfertility is the smoothed version of Australiafertility data. The smoothing technique is the penalized regression spline with concave constraint, described in Hyndman and Ullah (2007).

Author(s)

Han Lin Shang

Source

The Australian Demographic Data Bank (courtesy of Len Smith).

References

R. J. Hyndman and M. S. Ullah (2007) "Robust forecasting of mortality and fertility rates: A functional data approach", *Computational Statistics and Data Analysis*, **51**(10), 4942-4956.

R. J. Hyndman and H. Booth (2008) "Stochastic population forecasts using functional data models for mortality, fertility and migration", *International Journal of Forecasting*, **24**(3), 323-342.

R. J. Hyndman and H. L. Shang (2009) "Forecasting functional time series (with discussion)", *Journal of the Korean Statistical Society*, **38**(3), 199-221.

Examples

```
plot(Australiafertility)
plot(Australiasmoothfertility)
```

ElNino — *Sea surface temperature data set from January 1950 to December 2006*

Description

Original monthly sea surface temperatures have been restricted from January 1950 to December 2006.

The monthly sea surface temperatures can be smoothed using smoothing spline with the smoothing parametric determined by generalized cross validation.

Usage

```
data(ElNino)
data(ElNinosmooth)
```

Format

An object of class `sfts`.

Details

These averaged monthly sea surface temperatures are measured by the different moored buoys in the "Nino region" defined by the coordinates 0-10 degree South and 90-80 degree West.

Source

National Weather Service Climate Prediction Center website at `http://www.cpc.ncep.noaa.gov/data/indices/sstoi.indices`. The data is the third column with the title NINO1+2.

This data set can also be found at the NonParametric Functional Data Analysis website (`http://www.lsp.ups-tlse.fr/staph/npfda/`).

References

A. Antoniadis and T. Sapatinas (2003) "Wavelet methods for continuous-time prediction using Hilbert-valued autoregressive processes", *Journal of Multivariate Analysis*, **87**(1), 133-158.

P. C. Besse, H. Cardot and D. B. Stephenson (2000) "Autoregressive forecasting of some functional climatic variations", *Scandinavian Journal of Statistics*, **27**(4), 673-687.

F. Ferraty, A. Rabhi and P. Vieu (2005) "Conditional quantiles for dependent functional data with application to the climate EL Nino Phenomenon", *Sankhya: The Indian Journal of Statistics*, **67**(2), 378-398.

F. Ferraty and P. Vieu (2007) *Nonparametric functional data analysis*, New York: Springer.

R. J. Hyndman and H. L. Shang (2010) "Rainbow plots, bagplots, and boxplots for functional data", *Journal of Computational and Graphical Statistics*, **19**(1), 29-45.

E. Moran, R. Adams, B. Bakoyema, S. Fiorini and B. Boucek (2006) "Human strategies for coping with El Nino related drought in Amazonia", *Climatic Change*, **77**(3-4), 343-361.

A. Timmermann, J. Oberhuber, A. Bacher, M. Esch, M. Latif and E. Roeckner (1999) "Increased El Nino frequency in a climate model forced by future greenhouse warming", *Nature*, **398**(6729), 694-697.

Examples

```
plot(ElNino)
plot(ElNinosmooth)
```

`ElNino2011` *Sea surface temperature data set from January 1950 to December 2011*

Description

Original monthly sea surface temperatures have been restricted from January 1950 to December 2011.

The monthly sea surface tmeperatures can be smoothed using smoothing spline with the smoothing parametric determined by generalized cross validation.

Usage

```
data(ElNino2011)
data(ElNino2011smooth)
```

Format

An object of class `sfts`.

Details

These averaged monthly sea surface temperatures are measured by the different moored buoys in the "Nino region" defined by the coordinates 0-10 degree South and 90-80 degree West.

Source

National Weather Service Climate Prediction Center website at `http://www.cpc.ncep.noaa.gov/data/indices/sstoi.indices`. The data is the third column with the title NINO1+2.

This data set can also be found at the NonParametric Functional Data Analysis website (`http://www.lsp.ups-tlse.fr/staph/npfda/`).

References

A. Antoniadis and T. Sapatinas (2003) "Wavelet methods for continuous-time prediction using Hilbert-valued autoregressive processes", *Journal of Multivariate Analysis*, **87**(1), 133-158.

P. C. Besse, H. Cardot and D. B. Stephenson (2000) "Autoregressive forecasting of some functional climatic variations", *Scandinavian Journal of Statistics*, **27**(4), 673-687.

F. Ferraty, A. Rabhi and P. Vieu (2005) "Conditional quantiles for dependent functional data with application to the climate EL Nino Phenomenon", *Sankhya: The Indian Journal of Statistics*, **67**(2), 378-398.

F. Ferraty and P. Vieu (2007) *Nonparametric functional data analysis*, New York: Springer.

R. J. Hyndman and H. L. Shang (2010) "Rainbow plots, bagplots, and boxplots for functional data", *Journal of Computational and Graphical Statistics*, **19**(1), 29-45.

E. Moran, R. Adams, B. Bakoyema, S. Fiorini and B. Boucek (2006) "Human strategies for coping with El Nino related drought in Amazonia", *Climatic Change*, **77**(3-4), 343-361.

A. Timmermann, J. Oberhuber, A. Bacher, M. Esch, M. Latif and E. Roeckner (1999) "Increased El Nino frequency in a climate model forced by future greenhouse warming", *Nature*, **398**(6729), 694-697.

Examples

```
plot(ElNino2011)
plot(ElNino2011smooth)
```

`fboxplot` *Functional bagplot and functional HDR boxplot*

Description

Compute bivariate bagplot, functional bagplot and bivariate HDR boxplot, functional HDR boxplot.

Usage

```
fboxplot(data, plot.type = c("functional", "bivariate"),type = c("bag",
 "hdr"), alpha = c(0.01, 0.5), factor = 2.57, na.rm = TRUE,
  xlab = data$xname, ylab = data$yname, shadecols = gray((9:1)/10),
   pointcol = 1, plotlegend = TRUE, legendpos = "topright", ncol = 2,
    ...)
```

Arguments

`data`	An object of class `fds` or `fts`.
`plot.type`	Version of boxplot. When `plot.type="functional"`, a functional plot is provided. When `plot.type="bivariate"`, a square bivariate plot is provided.
`type`	Type of boxplot. When `type="bag"`, a bagplot is provided. When `type="hdr"`, a HDR boxplot is provided.
`alpha`	Coverage probability for the functional HDR boxplot. α are the coverage percentages of the outliers and the central region.
`factor`	When `type="bag"`, the outer region of a bagplot is the convex hull obtained by inflating the inner region by the bagplot factor.
`na.rm`	Remove missing values.
`xlab`	A title for the x axis.
`ylab`	A title for the y axis.
`shadecols`	Colors for shaded regions.
`pointcol`	Color for outliers and mode.
`plotlegend`	Add a legend to the graph.
`legendpos`	Legend position. By default, it is the top right corner.
`ncol`	Number of columns in the legend.
`...`	Other arguments.

Details

The functional curves are first projected into a finite dimensional subspace. For simiplicity, we choose the subspace as R^2. Based on Tukey (1974)'s halfspace bagplot and Hyndman (1996)'s HDR boxplot, we order each data point in R^2 by data depth and data density. Outliers are those that have either lowest depth or lowest density.

Value

Function produces a plot.

Author(s)

Rob J Hyndman, Han Lin Shang

References

J. W. Tukey (1974) "Mathematics and the picturing of data", *Proceedings of the International Congress of Mathematicians*, **2**, 523-532, Canadian Mathematical Congress, Montreal.

P. Rousseeuw, I. Ruts and J. Tukey (1999) "The bagplot: A bivariate boxplot", *The American Statistician*, **53**(4), 382-387.

R. J. Hyndman (1996) "Computing and graphing highest density regions", *The American Statistician*, **50**(2), 120-126.

R. J. Hyndman and H. L. Shang. (2010) "Rainbow plots, bagplots, and boxplots for functional data", *Journal of Computational and Graphical Statistics*, **19**(1), 29-45.

See Also

SVDplot

Examples

```
fboxplot(data = ElNino, plot.type = "functional", type = "bag")
fboxplot(data = ElNino, plot.type = "bivariate", type = "bag")
fboxplot(data = ElNino, plot.type = "functional", type = "hdr", alpha = c(0.07,0.5))
fboxplot(data = ElNino, plot.type = "bivariate", type = "hdr", alpha = c(0.07,0.5))
```

---

fdepth *Compute functional depth.*

Description

Compute functional depth.

Usage

```
fdepth(data, type = c("FM", "mode", "RP", "RPD"), trim = 0.25)
```

Arguments

`data` An object of class `fds` or `fts`.

`type` Type of functional depth.

`trim` Percentage of trimming.

Details

If `type="FM"`, it computes the functional depth of Fraiman and Muniz (2001), which is considered as the first functional depth.

If `type="mode"`, it computes the functional depth of Cuevas et al. (2006). A functional mode is defined as the curve most densely surrounded by the rest of curves of the dataset.

If `type="RP"` and `type="RPD"`, it computes random projection functional depth of Cuevas et al. (2007). Cuevas et al. (2007) considered the random projection depth based on measuring the depth of the functional data under projections and taking additional information of their derivatives. The basic idea is to project each functional curve, along a random direction, defining a point in R^2. A data depth in R^2 provides an order of the projected points.

The argument trim=0.25 first order curves by depth, and then trim 25 percent curves that have comparably lower depth.

Value

A list containing the following components is returned.

`median` Median curve (highest depth).

`lmed` Index of median curve.

`ltrim` Indexes of the trimmed curves.

`prof` Functional depth for each curve.

`mtrim` Mean of trimmed curves.

Author(s)

Han Lin Shang

References

A. Cuevas and M. Febrero and R. Fraiman (2001) "Cluster Analysis:a further approach based on density estimation", *Computational Statistics \& Data Analysis*, **36**(4), 441-456.

A. Cuevas and M. Febrero and R. Fraiman (2006) "On the use of bootstrap for estimating functions with functional data", *Computational Statistics \& Data Analysis*, **51**(10), 1063-1074.

A. Cuevas and M. Febrero and R. Fraiman (2007) "Robust estimation and classification for functional data via projection-based depth notions", *Computational Statistics*, **22**(3), 481-496.

R. Fraiman and G. Muniz (2001) "Trimmed means for functional data", *Test*, **10**(2), 419-440.

M. Febrero and P. Galeano and W. Gonzalez-Manteiga (2008) "Outlier detection in functional data by depth measures, with application to identify abnormal NOx levels", *Environmetrics*, **19**(4), 331-345.

Examples

```
fdepth(data = ElNino, type = "FM")
fdepth(data = ElNino, type = "mode")
fdepth(data = ElNino, type = "RP")
fdepth(data = ElNino, type = "RPD")
```

---

fds *Create functional objects*

Description

The function `fds` is used to create general functional objects that are not ordered by time. The function `fts` is used to create functional time series objects. The function `sfts` is used to create sliced functional time series objects, where the `x` variable is also a time variable.

Usage

```
fds(x, y, xname, yname)
fts(x, y, start = 1, frequency = 1, xname, yname)
sfts(data, period = frequency(data), start = tsp(data)[1],
     frequency = 1, xname, yname)
```

Arguments

`x`	Numeric vector of length p.
`y`	Matrix of size $p \times n$ representing n functions of x observed at points $1, \ldots, p$.
`data`	An object of class `ts`.
`period`	Time period of sliced functional data. For instance, `period = 12` is a monthly data.
`start`	The time of the first observation. Either a single number or a vector of two integers, which specify a natural time unit and a (1-based) number of samples into the time unit. See `ts` for details.
`frequency`	The number of observations per unit of time.
`xname`	Character string giving name of `x` vector. (optional)
`yname`	Character string giving name of `y` vector. (optional)

Value

An object of class `fds` or `fts` or `sfts`.

Author(s)

Rob J Hyndman

Examples

```
fds(x = 1:20, y = Simulationdata$y, xname = "x", yname = "Simulated value")
fts(x = 15:49, y = Australiasmoothfertility$y, xname = "Age",
    yname = "Fertility rate")
sfts(ts(as.numeric(ElNino$y), frequency = 12), xname = "Month",
     yname = "Sea surface temperature")
```

---

foutliers — *Functional outlier detection methods.*

Description

Functional outlier detection methods.

Usage

```
foutliers(data, method = c("robMah", "lrt", "depth.trim", "depth.pond",
 "HUoutliers"), dfunc = depth.mode, nb = 200, suav = 0.05, trim = 0.1,
  order = 2, lambda = 3.29,...)
```

Arguments

`data`	An object of class `fds` or `fts`.
`method`	Outlier detection method.
`dfunc`	When `method="lrt"` or `method="depth.trim"` or `method="depth.pond"`, users can specify the type of depth functions with possible choices of depth.FM, depth.mode, depth.RP, depth.RPD.
`nb`	When `method="lrt"`, users can specify the number of bootstrap samples.
`suav`	When `method="lrt"`, users can specify the smoothing parameter used in the smoothed bootstrap samples to determine the cutoff value.
`trim`	When `method="lrt"` or `method="depth.trim"` or `method="depth.pond"`, users can specify the trimming percentage.
`order`	When `method="HUoutliers"`, users can specify the number of principal components.
`lambda`	When `method="HUoutliers"`, users can specify the value of tuning parameter.
`...`	Other arguments.

Details

When `method="lrt"`, the outlier detection method corresponds to the approach of Febrero et al. (2007) using the likelihood ratio test.

When `method="depth.trim"`, the outlier detection method corresponds to the approach of Febrero et al. (2008) using the functional depth with trimmed curves.

When `method="depth.pond"`, the outlier detection method corresponds to the approach of Febrero et al. (2008) using the functional depth with all curves.

When `method="HUoutliers"`, the outlier detection method corresponds to the approach of Hyndman and Ullah (2008) using the integrated square forecast errors.

When `method="robMah"`, the outlier detection method corresponds to the approach of Rousseeuw and Leroy (1987) using the robust Mahalanobis distance.

Value

A list containing the following components is returned.

`outliers`	Detected outliers.
`cutoff`	Threshold value to separate outliers from non-outliers, when `method="lrt"`, `method="depth.trim"`, and `method="depth.pond"`.
`depth.total`	Depth measure of each functional curve.
`depth.out`	Depth measure of functional outliers.

Author(s)

Han Lin Shang

References

P. Rousseeuw and A. Leroy (1987) *Robust regression and outlier detection*, John Wiley and Sons, New York.

A. Atkinson (1994) "Fast very robust methods for the detection of multiple outliers", *Journal of the American Statistical Association*, **89**(428), 1329-1339.

R. J. Hyndman and M. S. Ullah (2007) "Robust forecasting of mortality and fertility rates: A functional data approach", *Computational Statistics and Data Analysis*, **51**(10), 4942-4956.

M. Febrero and P. Galeano and W. Gonzalez-Manteiga (2007) "A functional analysis of NOx levels: location and scale estimation and outlier detection", *Computational Statistics*, **22**(3), 411-427.

M. Febrero and P. Galeano and W. Gonzalez-Manteiga (2008) "Outlier detection in functional data by depth measures, with application to identify abnormal NOx levels", *Environmetrics*, **19**(4), 331-345.

R. J. Hyndman and H. L. Shang. (2010) "Rainbow plots, bagplots, and boxplots for functional data", *Journal of Computational and Graphical Statistics*, **19**(1), 29-45.

Examples

```
foutliers(data = ElNino, method = "lrt")
foutliers(data = ElNino, method = "depth.trim")
foutliers(data = ElNino, method = "depth.pond")
foutliers(data = ElNino, method = "HUoutliers")
foutliers(data = ElNino, method = "robMah")
```

plot.fdepth *Plot functional depth*

Description

Plot functional depth.

Usage

```
## S3 method for class 'fdepth'
plot(x, show.legend = TRUE, pos.legend = "bottomleft", ...)
```

Arguments

x	An object of class `fdepth`.
show.legend	Is legend required?
pos.legend	When `show.legend = TRUE`, users can specify the position of the legend.
...	Other plotting parameters passed to `par`.

Value

Function produces a plot.

Author(s)

Rob J Hyndman, Han Lin Shang

References

R. J. Hyndman and H. L. Shang. (2010) "Rainbow plots, bagplots, and boxplots for functional data", *Journal of Computational and Graphical Statistics*, **19**(1), 29-45.

See Also

`fdepth`

Examples

```
plot(fdepth(ElNino))
```

plot.fds *Plot functional objects*

Description

Plot functional curves.

Usage

```
## S3 method for class 'fds'
plot(x, plot.type = c("functions", "time", "depth", "density"),
  col = NULL, type = "l", lty = 1, xlab = x$xname, ylab = x$yname,
  pch = c(1:9,0, letters, LETTERS), add = FALSE, index,
  colorchoice = c("rainbow", "heat.colors", "terrain.colors",
  "topo.colors", "cm.colors", "rainbow_hcl", "gray", "sequential_hcl",
  "heat_hcl", "terrain_hcl", "diverge_hcl"), plotlegend = FALSE,
  legendpos = "topright", ncol = 1, ...)
## S3 method for class 'fds'
lines(x, type = "l", index, ...)
## S3 method for class 'fds'
points(x, type = "p", index, ...)
```

Arguments

x	An object of class fds or fts.
plot.type	Type of plot. See details for more explanations.
col	Colors to use in plot. Default in plot.fds is to use a rainbow color palette with the number of colors equal to the number of functions.
type	1-character string giving the type of plot desired.
lty	The line type.
xlab	A title for x axis.
ylab	A title for y axis.
pch	Either an integer specifying a symbol or a single character to be used as the default in plotting points.
add	If add = TRUE, it plots a line or points.
index	Index of a specific curve that is plotted as a line or points.
colorchoice	Color palette used for drawing the rainbow plot.
plotlegend	Add a legend to the graph.
legendpos	Position of legend.
ncol	Number of column in the legend.
...	Other plotting parameters passed to par.

Details

If `plot.type="functions"`, then functions are plotted using a rainbow color palette so the first few functions are shown in red, followed by orange, yellow, green, blue and indigo with the last few functions plotted in violet.

If `plot.type="time"`, then each value of x is shown as a separate time series in a time plot.

If `plot.type="depth"`, then functions are first ordered by depth and then plotted using a rainbow color palette.

If `plot.type="density"`, then functions are first ordered by density and then plotted using a rainbow color palette.

Value

Function produces a plot.

Author(s)

Rob J Hyndman, Han Lin Shang

References

R. J. Hyndman and H. L. Shang. (2010) "Rainbow plots, bagplots, and boxplots for functional data", *Journal of Computational and Graphical Statistics*, **19**(1), 29-45.

See Also

`fds`, `lines.fds`, `points.fds`

Examples

```
plot(x = ElNinosmooth, plot.type = "time")
plot(x = ElNinosmooth, plot.type = "depth", legend = TRUE)
plot(x = ElNinosmooth, plot.type = "density", legend = TRUE)
plot(x = ElNinosmooth, plot.type = "functions", legend = TRUE)
lines(x = ElNinosmooth, plot.type = "functions", index = 3)
points(x = ElNinosmooth, plot.type = "functions", index = 3)
```

`Simulationdata` *Simulated data*

Description

Simulated data used in Hyndman and Shang (2008).

Usage

```
data(Simulationdata)
```

Format

An object of class fds.

References

R. J. Hyndman and H. L. Shang. (2010) "Rainbow plots, bagplots, and boxplots for functional data", *Journal of Computational and Graphical Statistics*, **19**(1), 29-45.

Examples

```
plot(Simulationdata, col = rainbow(100))
lines(Simulationdata, index = 991:1000, col = "black")
```

---

summaryfunction — *Summary statistics for functional data*

Description

Display summary statistics (minimum, 1st quantile, median, mean, 3rd quantile, maximum) and quantiles of functional data

Usage

```
summaryfunction(ftsdata, plot.type = c("summarystats", "quantilestats"),
 quantilepercent = seq(0.1, 0.9, by = 0.1), plot.legend = FALSE,
  legendpos = "topright", cex = 0.9, lwd = 1, lty = 1, ncol = 2)
```

Arguments

`ftsdata`	An object of class fds.
`plot.type`	Summary statistics or quantiles.
`quantilepercent`	Percentage of quantiles.
`plot.legend`	Plot the legend.
`legendpos`	Position of the legend.
`cex`	Point size.
`lwd`	Width of line.
`lty`	Line type.
`ncol`	Number of columns in the legend.

Details

A function for displaying summary statistics or quantiles of functional data.

Value

Return a plot of summary statistics of functional data or a plot of quantiles of functional data.

Author(s)

Han Lin Shang

See Also

fds

Examples

```
summaryfunction(Australiasmoothfertility, plot.type="summarystats")
summaryfunction(Australiasmoothfertility, plot.type="quantilestats",plot.legend = TRUE)
```

SVDplot *Singular value decomposition plot*

Description

The singular value decomposition (SVD) plot of Zhang et al. (2007) captures the changes in the singular columns as the number of curves gradually increases.

Usage

```
SVDplot(object, order = 3, plot = TRUE, plot.type = c("fts", "image"),
        mfrow = c(2, 3))
```

Arguments

object	An object of fds.
order	Number of SVD components. The maximum order is 4.
plot	Is graphical display required?
plot.type	Plot functional time series or images?
mfrow	Grid of graphics.

Details

By using the SVD, Zhang et al. (2007) proposed a plot for visualizing patterns of functional time series. They considered a set of curves as a two-way (p * n) data matrix, where p is the total number of covariates and n is the total number of curves.

The main advantage of this dynamic plot is to visualize both column and row information of a two-way matrix simultaneously, relate the matrix to the corresponding curves, show local variations, and highlight interactions between columns and rows of a two-way matrix.

Value

When `plot = TRUE`, it returns a plot.

When `plot = FALSE`, it returns the following:

`svds`	A number of singular value decomposition ordered by the amount of explained variation.
`reconstruction`	Reconstruction of the original data using the SVD.
`residual`	Residual of the original data.

Note

MATLAB code is available at `http://www.unc.edu/~lszhang/research/network/SVDmovie/`.

Using the animate package of Grahn(2011), a set of dynamic movies can be created to visualize the changes in singular rows and singular columns.

Author(s)

Han Lin Shang

References

L. Zhang, J. Marron, H. Shen and Z. Zhu (2007) "Singular value decomposition and its visualization", *Journal of Computational and Graphical Statistics*, **16**(4), 833-854.

A. Grahn (2011) "The animate Package", `http://ctan.unsw.edu.au/macros/latex/contrib/animate/animate.pdf`.

See Also

`fboxplot`, `svd`

Examples

```
SVDplot(ElNinosmooth)
SVDplot(ElNinosmooth, plot.type = "image")
```

Index